

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平5-100864

(43) 公開日 平成5年(1993)4月23日

(51) Int.Cl. <sup>8</sup>	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 9/45				
9/22	3 2 0 Z	9193-5B		
		9292-5B	G 0 6 F 9/44	3 2 0 D

審査請求 未請求 請求項の数1(全4頁)

(21) 出願番号 特願平3-262463

(22) 出願日 平成3年(1991)10月11日

(71) 出願人 000005821

松下電器産業株式会社

大阪府門真市大字門真1006番地

(72) 発明者 中村 彩男

大阪府門真市大字門真1006番地 松下電器  
産業株式会社内

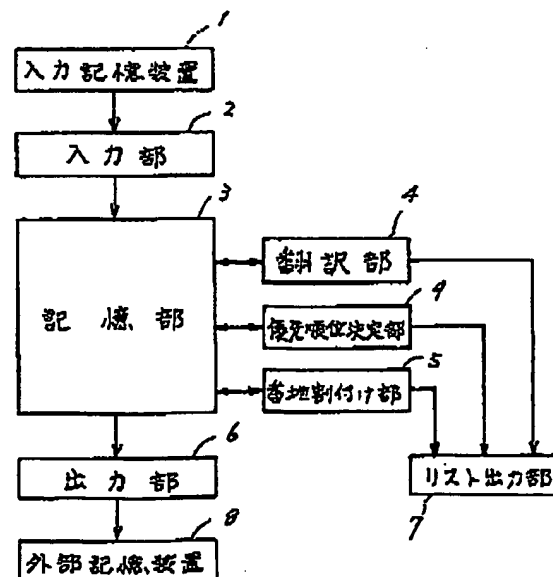
(74) 代理人 弁理士 森本 義弘

(54) 【発明の名称】 マイクロプログラムアセンブラ

(57) 【要約】

【目的】 プロセッサの実行時間に対する制御メモリの読出し待ち時間を最小にできるマイクロプログラムアセンブラを提供する。

【構成】 マイクロプログラムアセンブラにおいて、機械語に翻訳しているフォーマットのラベル参照テーブルを見て、ラベル参照テーブル上のフォーマットの、次に実行すべき番地の「ラベル部」の内容を、最も優先度の高いマイクロ命令の「ラベル部」の内容に座換する優先順位決定部9を設けたことにより、ソースコードをアセンブルすると、翻訳された各マイクロ命令の次に実行すべき番地の「ラベル部」には、最も優先度の高いマイクロ命令の制御メモリの物理アドレスがセットされる。これにより、プロセッサの命令読出しの待ち時間が少なくなり、実効的にプロセッサのサイクルタイムでプログラムを実行し、データ処理装置のパフォーマンスを向上することができる。



## 【特許請求の範囲】

【請求項1】 少なくとも2つ以上のラベル部と、命令部とを備えたフォーマットで記述されたマイクロ命令を機械語に変換するマイクロプログラムアセンブラであって、

前記2つ以上のラベル部の内、次に実行すべき番地のラベル部の内容を、優先度の高いマイクロ命令のラベル内容に変換する手段を備えたことを特徴とするマイクロプログラムアセンブラ。

## 【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、マイクロプログラムアセンブラ、特にそのマイクロ命令フォーマットの次に実行すべき番地のラベル部を変換する手段に関するものである。

【0002】

【従来の技術】 以下、従来のマイクロプログラムアセンブラの一例について説明する。マイクロプログラムをマイクロアセンブラ言語で記述して、機械語にアセンブルする場合、アセンブル言語で記述される各マイクロ命令のフォーマットは、「第1ラベル部」、「第2ラベル部」、「命令部」に区切られる。なお、「ラベル部」は2つ以上になることもある。

【0003】 ここで、「第1ラベル部」：番地につける名標、「第2ラベル部」：次に実行すべき番地のラベル、「命令部」：処理内容を指定するコード、である。

【0004】 水平型マイクロプログラムに対するアセンブラは、上記形式で記述されたフォーマットを読み出し、機械語に翻訳する。図2に従来のマイクロプログラムアセンブラの構成図を示す。

【0005】 図2において、1はマイクロプログラムのソースコードが格納されている入力記憶装置であり、この入力記憶装置1からソースコードは入力部2を通じて記憶部3に格納される。ソースコードは記憶部3に格納されると、翻訳部4において、各マイクロ命令のフォーマットを読み出し、「命令部」の命令ニーモニックを機械語に変換し、リスト出力部7に出力すると共に、機械語に翻訳しているフォーマットが、どのフォーマットの「ラベル部」を指定しているかを示すラベル参照テーブルを作成して記憶部3に格納する。次に、番地割付け部5においては、各マイクロ命令に対して制御メモリ上の物理番地を割り付け、リスト出力部7に出力すると共に、割付け結果の番地を、各マイクロ命令の「アドレス部」にセットする。

【0006】 この様にして、ソースコードの各フォーマットを「命令部」、「アドレス部」からなる機械語に翻訳した結果は、出力部6を介して外部記憶装置8に出力され、オブジェクトコードとして格納される。

【0007】 実行に際して、機械語に翻訳されたマイ

クロプログラムは、外部記憶装置8から制御メモリに格納され、制御メモリのアクセスタイムにしたがって読み出され、実行される。

【0008】

【発明が解決しようとする課題】 しかしながら上記の従来の構成では、制御メモリのアクセスタイムがプロセッサのサイクルタイムより遅い場合、プロセッサの実行時間に対して待ち時間が必要になり、プロセッサの処理能力を低下させるという問題を有していた。

【0009】 本発明は上記問題を解決するものであり、プロセッサ実行時間に対する待ち時間を最小にできるマイクロプログラムアセンブラを提供することを目的とするものである。

【0010】

【課題を解決するための手段】 上記問題を解決するため本発明のマイクロプログラムアセンブラは、ラベル部の内、次に実行すべき番地のラベル部の内容を、優先度の高いマイクロ命令のラベル内容に変換する手段を備えた構成を有している。

【0011】

【作用】 上記構成により、翻訳された各マイクロ命令の次に実行すべき番地のラベル部には、最も優先度の高いマイクロ命令の制御メモリの物理アドレスがセットされ、高速にアクセスされる。よって、命令読みだしのための待ち時間が少なくなり、実効的にプロセッサのサイクルタイムでプログラムを実行することが可能になる。

【0012】

【実施例】 以下、本発明の一実施例を図面に基づいて説明する。なお、従来例の図2と同一の構成には同一の符号を付して説明を省略する。

【0013】 図1において、9は予めラベルの順位が設定された優先順位決定部であり、機械語に翻訳しているフォーマットのラベル参照テーブルを見て、ラベル参照テーブル上のフォーマットの「第2ラベル部」の内容を、最も優先度の高いマイクロ命令の「ラベル部」の内容に変換する。

【0014】 以上のように構成されたマイクロプログラムアセンブラについて、以下その動作について説明する。まず、ソースコードが入力記憶装置1から入力部2を通じて記憶部3に格納される。ソースコードが記憶部3に格納されると、翻訳部4において、各フォーマットを読み出し、「命令部」の命令ニーモニックを機械語に変換し、リスト出力部7に出力すると共に、ラベル参照テーブルを作成して記憶部3に格納する。

【0015】 ここで優先順位決定部9を設けることにより、現在機械語に翻訳しているフォーマットが、どのフォーマットの「ラベル部」を指定しているかを示すラベル参照テーブルを見て、最も優先度の高いマイクロ命令の「ラベル部」の内容を認識し、ラベル参照テーブル上のフォーマットの「第2ラベル部」に挿入する事ができ

る。また、変換した「ラベル部」の内容はリスト出力部7に出力される。

【0016】次に、番地割付け部5においては、各マイクロ命令に対して制御メモリ上の物理番地を割り付け、リスト出力部7に出力すると共に、割付け結果の番地を、各マイクロ命令の「アドレス部」にセットする。

【0017】このようにして、ソースコードの各フォーマットを「命令部」、「アドレス部」からなる機械語に翻訳した結果を外部記憶装置8に、オブジェクトコードとして格納する。

【0018】以上のように本実施例によれば、マイクロプログラムアセンブルにおいて、ソースコードをアセンブルすると、翻訳された各マイクロ命令の「第2ラベル部」には、最も優先度の高いマイクロ命令の制御メモリの物理アドレスがセットされ、高速なアクセスを実行することができ、命令読みだしのための待ち時間が少なくなり、実効的にプロセッサのサイクルタイムでプログラムを実行することができる。

【0019】

【発明の効果】以上のように本発明によれば、翻訳された各マイクロ命令の次に実行すべき番地のラベル部には、最も優先度の高いマイクロ命令の制御メモリの物理

アドレスがセットされることにより、高速にアクセスすることができ、命令読みだしのための待ち時間が少なくなり、実効的にプロセッサのサイクルタイムでプログラムを実行することができ、データ処理装置のパフォーマンスを向上することができる優れたマイクロプログラムアセンブラを提供することができる。

【図面の簡単な説明】

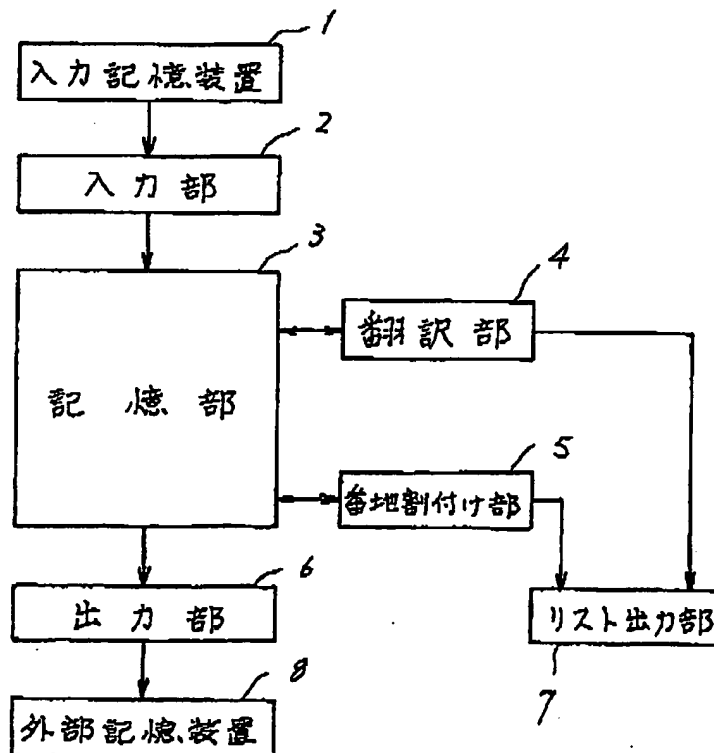
【図1】本発明の一実施例におけるマイクロプログラムアセンブラの構成図である。

10 【図2】従来のマイクロプログラムアセンブラの構成図である。

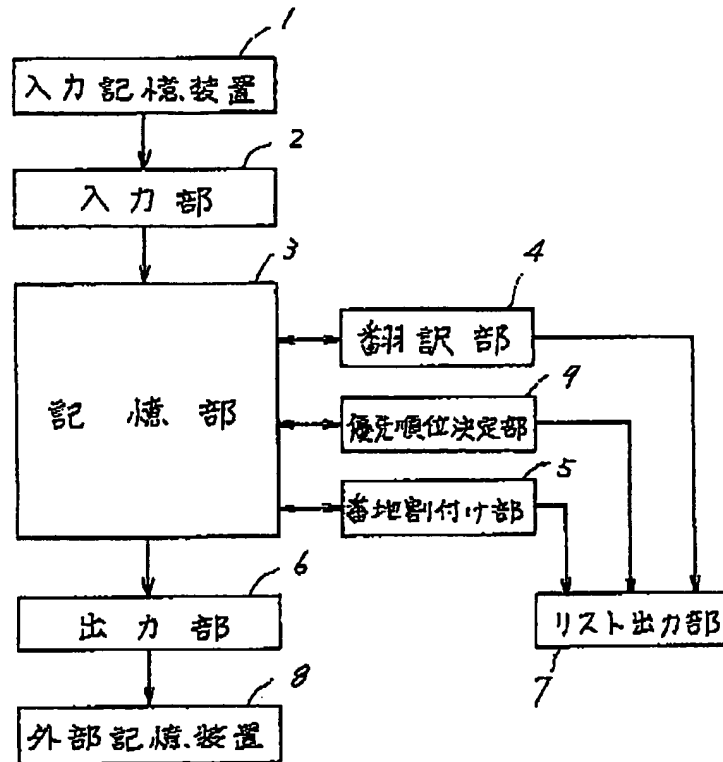
【符号の説明】

- 1 入力記憶装置
- 2 入力部
- 3 記憶部
- 4 翻訳部
- 5 番地割付け部
- 6 出力部
- 7 リスト出力部
- 8 外部記憶装置
- 9 優先順位決定部

【図2】



【図1】



Laid-open Number: Hei 5-100864  
Laid-open Date: April 23, 1993  
Application Number: Hei 3-262463  
Filing Date: October 11, 1991

(54) [Title of the Invention] Microprogram assembler

(57) [Abstract]

[Object] To provide a microprogram assembler capable of minimizing a waiting time for control-memory reading with respect to an execution time of a processor.

[Construction] In a microprogram assembler, there is provided a priority determination section 9 which refers to a label reference table of a format being translated into a machine language to convert a content of a "label part" of an address to be executed next, of the format in the label reference table, into a content of the "label part" of a highest-priority microinstruction. Therefore, when a source code is assembled, a physical address on a control memory of the highest-priority microinstruction is set in the "label part" of the address to be executed next of each of translated microinstructions. Thus, a waiting time for a processor to read out an instruction is reduced and a program is effectively executed in a cycle time of the processor to improve performance of a data processing device.

[Scope of Claim]

[Claim 1] A microprogram assembler for converting a microinstruction written in a format having at least two label parts and an instruction part into a machine language, the microprogram assembler being characterized by comprising:

means for converting a content of a label part of an address to be executed next, the label part being one of the at least two label parts, into a content of a label of a high-priority microinstruction.

[Detailed Description of the Invention]

[0001]

[Field of the Industrial Application] The present invention relates to a microprogram assembler, and more particularly to means for converting a label part of an address to be executed next, in a microinstruction format.

[0002]

[Prior Art] Hereinafter, examples of the conventional microprogram assemblers will be described. When a microprogram is written in a micro-assembler language and is assembled into a machine language, a format of each microinstruction written in an assemble language has divisions of a "first label part", a "second label part", and an "instruction part". Note that two or more "label parts" may be included.

[0003] Here, the "first label part" shows a name for the address,

the "second label part" shows a label of the address to be executed next, and the "instruction part" shows a code that specifies a content of processing.

[0004] An assembler for horizontal microprograms reads out a format written in the above-mentioned manner and translates it into the machine language. Fig. 2 shows a configuration diagram of the conventional microprogram assembler.

[0005] In Fig. 2, reference numeral 1 designates an input storage device in which a source code of a microprogram is stored. The source code from the input storage device 1 is stored in a storage section 3 via an input section 2. When the source code is stored in the storage section 3, a translation section 4 reads out a format of each microinstruction, translates an instruction mnemonic in the "instruction part" into the machine language, outputs to a list output section 7, and also creates a label reference table indicating that the "label part" of which format is specified by the format being translated into the machine language and stores the table in the storage section 3. Next, an address allocation section 5 allocates a physical address on a control memory to each microinstruction, outputs the address to the list output section 7, and also sets the address obtained by the result of the allocation in an "address part" of the microinstruction.

[0006] In this manner, each format of the source code is translated into the machine language consisting of the "instruction part" and

the "address part". The result of translation is output via an output section 6 to an external storage device 8, and stored as an object code.

[0007] Upon executing, the microprogram translated into the machine language is stored in the control memory from the external storage device 8, read out at an access time of the control memory, and executed.

[0008]

[Problem to be solved by the Invention] However, the above-mentioned conventional configuration has a problem in that, when the access time of the control memory is longer than the cycle time of the processor, a waiting time is required with respect to an execution time of the processor, decreasing a processing ability of the processor.

[0009] The present invention has been made to solve the above problem. An object of the present invention is to provide a microprogram assembler capable of minimizing a waiting time with respect to an execution time of a processor.

[0010]

[Means for solving the Problem] In order to solve the above problem, a microprogram assembler of the present invention has configuration which includes means for converting a content of a label part of an address to be executed next, the label part being one of label parts, into a content of the label of a high-priority

microinstruction.

[0011]

[Operation] With the above configuration, in the label part of the address to be executed next of each of the translated microinstruction, a physical address on the control memory of the highest-priority microinstructions is set to be accessed in a high speed. Therefore, a waiting time for reading out an instruction is reduced, making it possible to effectively execute a program in the cycle time of the processor.

[0012]

[Embodiment] Hereinafter, an embodiment of the present invention will be described based on the drawing. Note that the same reference numerals are given to the same configuration as in the conventional example shown in Fig. 2 and a description thereof is omitted.

[0013] In Fig. 1, reference numeral 9 designates a priority determination section in which an order of the label is set in advance. The priority determination section 9 refers to the label reference table of the format being translated into the machine language to convert the content of the "second label part" of the format in the label reference table into the content of the "label part" of the highest-priority microinstruction.

[0014] An operation of a microprogram assembler configured as described above will be described hereinafter. First, a source code from an input storage device 1 is stored in a storage section 3

via an input section 2. When the source code is stored in the storage section 3, a translation section 4 reads out each format, converts the instruction mnemonic of the "instruction part" into the machine language, outputs to a list output section 7, and also creates the label reference table and stores the table in the storage section 3.

[0015] Here, by providing the priority determination section 9, it is possible to refer to the label reference table indicating that the "label part" of which format is specified by the format now being translated into the machine language, to recognize the content of the "label part" of the highest-priority microinstruction, and to insert to the "second label part" of the format in the label reference table. Then, the converted content of the "label part" is output to the list output section 7.

[0016] Next, an address allocation section 5 allocates a physical address on the control memory to each microinstruction, outputs to the list output section 7, and also sets the address obtained by the result of the allocation in an "address part" of the microinstruction.

[0017] In this manner, each format of the source code is translated into the machine language consisting of the "instruction part" and the "address part". The result of translation is stored in an external storage device 8 as an object code.

[0018] According to the present embodiment, as described above,

in the microprogram assembler, when a source code is assembled, a physical address on the control memory of the highest-priority microinstruction is set in the "second label part" of the translated microinstruction. Thus, it is possible to perform a high-speed access, to reduce a waiting time for reading out an instruction, and to effectively execute a program in the cycle time of the processor.

[0019]

[Effect of the Invention] According to the present invention, as described above, a physical address on the control memory of the highest-priority microinstruction is set in the label part of the address to be executed next, of each of the translated microinstructions, thereby it is possible to access in a high speed, to reduce a waiting time for reading out an instruction, to effectively execute a program in the cycle time of the processor, and to provide a superior microprogram assembler capable of improving the performance of a data processing device.

[Brief Description of the Drawings]

[Fig. 1] A configuration diagram of a microprogram assembler according to one embodiment of the present invention.

[Fig. 2] A configuration diagram of a conventional microprogram assembler.

[Description of Reference Numerals]

1 input storage device

- 2 input section
- 3 storage section
- 4 translation section
- 5 address allocation section
- 6 output section
- 7 list output section
- 8 external storage device
- 9 priority determination section

【図1】

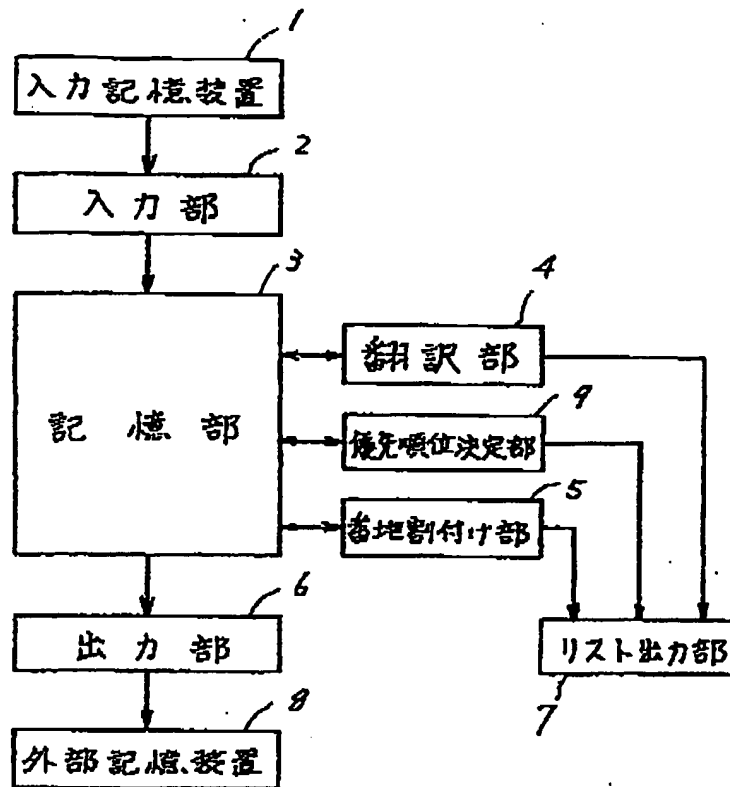


FIG. 1

- 1 INPUT STORAGE DEVICE
- 2 INPUT SECTION
- 3 STORAGE SECTION
- 4 TRANSLATION SECTION
- 5 ADDRESS ALLOCATION SECTION
- 6 OUTPUT SECTION
- 7 LIST OUTPUT SECTION
- 8 EXTERNAL STORAGE DEVICE
- 9 PRIORITY DETERMINATION SECTION

【図2】

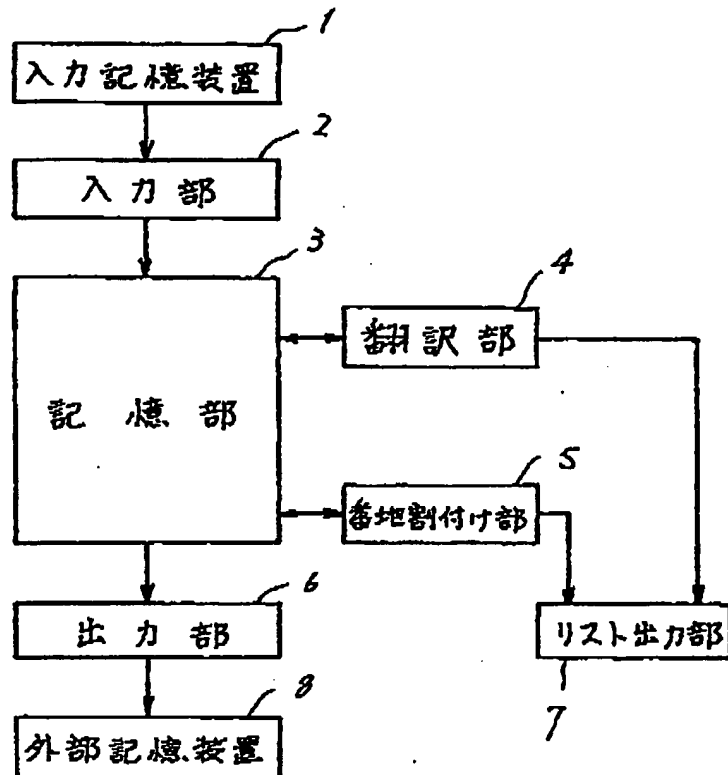


FIG. 2

1 INPUT STORAGE DEVICE

2 INPUT SECTION

3 STORAGE SECTION

4 TRANSLATION SECTION

5 ADDRESS ALLOCATION SECTION

6 OUTPUT SECTION

7 LIST OUTPUT SECTION

8 EXTERNAL STORAGE DEVICE